Vol.2021-CE-162 No.19
Vol.2021-CLE-35 No.19
2021/12/5

「情報関係基礎」プログラミング問題で省略されている変数初期化部分に関係するWaPEN，PyPENの実装

中西 渉[1,a)]

概要：大学入試センター試験，共通テストの数学②で選択できる「情報関係基礎」のプログラミング問題で用いられている擬似言語 DNCL は，実行環境が複数開発されており，テストのプログラムがそのまま実行できると言われることが多いが，実は問題で示されているプログラムはそのままでは実行できない．多くの場合データの初期化にあたる部分が省略されているので，実行するためにはその部分を補う必要がある．本稿では省略されている初期化部分を補う方法について考察し，筆者が開発している学習環境 WaPEN および PyPEN でそれを解決するために行った実装について述べる．

キーワード：プログラミング教育，大学入試，DNCL

Implementation of WaPEN and PyPEN for the omitted initialization part in the programming problems of “Informatics Basics”

Nakanishi Wataru[1,a)]

Abstract: Many learning environment for DNCL, pseudo programming language made for the University Entrance Examinations of National Center, was developed, and it is often said that program of the problems can be executed. But actually, they cannot run as is. In many cases, the part that initializes the data is omitted, so we must make up that part to execute it. In this paper, I consider about the way to complete those programs, and describe the implementation made to solve it in the learning environment WaPEN and PyPEN.

Keywords: Education of programming, University entrance examination, DNCL

1. はじめに

日本語プログラミング言語として広く知られている「なでしこ」[1] が大学入試共通テスト手順記述標準言語 DNCL に対応したと発表された [2]．といっても DNCL で書かれたプログラムがそのまま「なでしこ」で動くということではなく，命令や構文の対応表にしたがって書き換えることで，動かすことができるというものであったため，「なでしこ」そのものによる対応とはいえないのではないかとの印象も持たれたように思われる．

しかしそもそも大学入試センター試験や共通テスト（以下では「センター試験等」と表記する）「情報関係基礎」のプログラミング問題に掲載されている問題は，そのままでは動かないことが多い．いくつかの問題では掲載されたプログラムの前にデータの初期化が必要であるのにそれが記載されていないため，実行するには初期化のコードを補う必要がある．

本稿ではセンター試験等で出題されたプログラミング問題について，どのような初期化部分が省略されていたかについて調査を行い，筆者が開発している WaPEN，PyPEN でそれらを実現するために行った追加実装について述べる．

[1] 名古屋高等学校
Nagoya Senior High School
[a)] watayan@meigaku.ac.jp

ⓒ 2021 Information Processing Society of Japan

Vol.2021-CE-162 No.19
Vol.2021-CLE-35 No.19
2021/12/5

2. DNCL の概要

DNCL はセンター試験等で「情報関係基礎」のプログラミング問題の出題に使用されている，日本語を基本とした擬似言語である．言語仕様は Web で公開されており [3]，共通テストに合わせて更新された [4]．

変数は宣言せずに使うことができ，配列も使用できる．配列の添字が 0 から始まるか 1 から始まるかについては [3] には明確な記述はないが，たいていの問題は 1 から始まるものとして作られており，[4] では添字は 0 以上であるが 1 以上に限って使うこともあると記されている．制御構文は `if`〜`elif`〜`else` に相当する選択や，`while`〜`wend`，`do`〜`until`，`for`〜`next` に相当する繰り返し文がある．また，問題に応じて臨時で関数が作られることもある．

「情報 I」が使われる 2025 年度からの共通テストについて，大学入試センターから試作問題 [5]，サンプル問題 [6] が発表されており，そこでは Python 風の新しい DNCL が用いられているが，細かい仕様についてはまだ不明である．

3. 関連研究

3.1 DNCL 学習環境

DNCL を用いたプログラミング学習環境 PEN [7] および，それに筆者がフローチャートを付加した PenFlowchart [8] は Java アプリケーションなのでインストールが必要であり，また JRE がない環境では実行できない．そのため Web ブラウザ上の学習環境が複数作られた．

本多らによる「どんくり」[9] は DNCL を JavaScript にトランスパイルして実行している．DNCL に加え，C に似た英語表示のプログラムとの相互変換や変数一覧の確認，各構文が呼び出された回数や実行時間などを確認する機能が追加されている．大宮らによる wPEN [10] は短冊によるプログラミングができ，作問機能・解答機能を備えている．大門らによる Tetra [11] や XTetra [12] ではブロックでもタイピングでもプログラミングができるようになっており，実行時に各構文が実行された回数を可視化するなどの機能がある．

筆者は PenFlowchart を WaPEN [13] という形で Web ブラウザ上で動くように移植した．これはコードとフローチャートのどちらでもプログラミングが可能である（ただし，フローチャートは自作関数や自作手続きのあるプログラムには対応していない）．またその後，構文を Python 風にアレンジした PyPEN [14] を開発したところ，共通テストの試作問題やサンプル問題のプログラミングの問題で実際にそのような言語が使われていた．前述した DNCL の学習環境も，新しい DNCL に対応したものが開発されている [15–17]．

表 1 プログラミング問題の初期化に関する部分
Table 1 The initialization part of the programming problems.

年度	題材	省略	初期化内容
1998	ヒットアンドブロー	n	4 桁整数の各桁の数からなる配列
追試	Eight Queens	y	盤面を表す 2 次元配列
1999	配列のマージ	y	身長の値の配列
2000	探索	y	25 枚のカードの数の配列
2001	お釣りの金種計算	y	貨幣の枚数
2002	迷路	y	盤面を表す 2 次元配列
2003	文字列検索	y	文字列の各文字からなる配列
2004	小町算	n	記号を示す配列（-1 で初期化）
2005	じゃんけん	一部	手を表す番号の配列
2006	ライフゲーム	y	盤面を表す 2 次元配列
2007	計算ゲーム	y	盤面を表す 2 次元配列
2008	ソート（4 人）	n	名前と得点の配列
2009	素因数分解	n	割り切る数の配列（0 で初期化）
2010	漢数字への変換	n	使う漢字を入れた配列
2011	成績集計	一部	得点・順位の配列など
2012	分銅	n	分銅の枚数を入れる配列（0 で初期化）
2013	営業時間の最適化	y	時間帯ごとの利益の配列
2014	太鼓の楽譜	n	楽譜を表す配列
2015	ボードゲーム	y	盤面の数を入れる 2 次元配列
2016	荷物詰め込み	y	荷物の重さなどの配列
2017	三角形を数える	y	頂点間を結ぶ辺の名前の 2 次元配列
2018	迷路	y	盤面を表す 2 次元配列
2019	グループ分け	y	希望を表す 2 次元配列
2020	宝探しゲーム	y	罠の座標の配列
2021	すごろく	y	倍率の配列，位置の配列（長さ不定）
試作	シーザー暗号	y	暗号文の文字からなる配列
ｻﾝﾌﾟﾙ	ドント方式	n	党名・票数・当選者数の配列

3.2 センター試験等の問題で省略された部分

情報関係基礎のプログラミング問題のほとんどで配列が使われているが，その初期化については明示されていないことがよくある．情報処理学会情報入試委員会が公開している情報関係基礎アーカイブ [18] を参考に，センター試験や試作問題，サンプル問題でどのような配列の初期化が省略されているかを**表 1** に示した．表の「省略」カラムは，プログラムの実行に必要な初期化部分が省略されているかどうかを表している．

「どんくり」では 2008〜2018 年度のセンター試験等のプログラミングの問題を完成させたものを，サンプルプログラムとして紹介している [19]．中には表 1 では初期化部分の省略があるとしているものもあるが，プログラムとしては書かれていなくても問題文中で値が指定されていることも多いので，サンプルプログラムはその値を用いた初期化が追加されている．

ⓒ 2021 Information Processing Society of Japan

Vol.2021-CE-162 No.19
Vol.2021-CLE-35 No.19
2021/12/5

```
1:/* コンピュータとじゃんけんするプログラム */
2:
3:整数 a, x, i
4:文字列 te[2]
5:
6:te[0] ← "グー", te[1] ← "チョキ", te[2] ← "パー"
```

図 1　じゃんけんの手の初期化（PEN 版）
Fig. 1　Initialization of 'Janken' hand (using PEN).

```
1:/* コンピュータとじゃんけんをするプログラム */
2:
3:整数 a,x,i
4:文字列 te[2]
5:
6:te←{「グー」,「チョキ」,「パー」}
```

図 2　じゃんけんの手の初期化（PenFlowchart 版）
Fig. 2　Initialization of 'Janken' hand (using PenFlowchart).

```
a=[0,0,0,0,0,0,0,0,0,0]
b=10 個の 0
c=[0]*10
```

図 3　指定された長さの配列を初期化
Fig. 3　Initialization of array with specified length.

4. センター試験等のプログラミング問題における省略部分および WaPEN と PyPEN の追加実装

4.1 PyPEN から WaPEN への移植

最近筆者の開発が PyPEN に偏っており，新機能などは PyPEN では実装するものの WaPEN は放置されていた．そこで構文解析など言語の違いを除くほとんどの機能を，PyPEN から WaPEN に移植した．これにより PEN や PenFlowchart，WaPEN で動いていたプログラムが動かなくなるのだが，逆にそれによってセンター試験等の問題に近づいたと考えることもできる．たとえば変数の宣言をできなくしたことがそれにあたる．

4.2 長さの決まっている配列

PEN では**図 1** のように配列への値の代入が 1 つずつしかできなかったので，PenFlowchart では**図 2** のようにまとめて値を代入する構文を追加した．さらに WaPEN や PyPEN では**図 3** のような代入もできるようにした．なお，「《個数》個の《値》」と「[《値》]*《個数》」では，前者は《値》が《個数》回評価されるのに対し，後者は 1 回評価したものが繰り返し使われるという違いがある．たとえば「10 個の `random()`」は乱数値を 10 回生成するが，「`[random()]*10`」は 1 つの乱数値を 10 回繰り返すものになる．

4.3 長さの決まっていない配列

過去の出題では長さの決まっていない配列やそれに初期値を与えたものがよく使われているが，WaPEN や PyPEN ではこれは実装していない．PyPEN は Python で実行できるコードを出力できる点を維持したいので対応する考えはないが，DNCL の仕様にある構文なのだから WaPEN では実装したい．

「どんくり」はこのような配列の初期化に対応しており，値が代入された要素が参照されるときはその代入された値を返し，代入されていない要素が参照されるときにはその初期値を返すように実装されているので，WaPEN でもこれを参考にしたいと考えている．

```
/*2018年 情報関係基礎 第3問 問2*/
tate←9,yoko←11
Masu←{{1,1,1,1,1,1,1,1,1,1},
{1,0,0,0,1,0,0,0,1},
{1,1,1,0,0,0,1,0,1},
{9,0,0,0,1,0,1,0,1},
{1,1,1,0,1,0,1,0,1},
{1,0,1,1,1,0,1,1,1},
{1,0,0,0,1,0,1,0,1},
{1,1,1,0,0,0,1,0,9},
{1,0,0,0,1,1,1,0,1},
{1,0,1,0,0,0,0,0,1},
{1,1,1,1,1,1,1,1,1}}
```

図 4　2018 年度問題の初期化部分（Tetra 版）
Fig. 4　Initialization part of the problem in 2018 (using Tetra).

4.4 2 次元配列

たとえば 2018 年度センター試験第 3 問は迷路を解く問題であるため，最初にそのコースを 2 次元配列の形で与えなくてはいけないのだが，その部分は問題では省略されている．Tetra [20] は複数行にわたる代入文をサポートしているので，**図 4** のようにスマートな表記をすることができる．しかし WaPEN や PyPEN ではこのような複数行にわたる初期化を実装していなかった．そのため当時は**図 5** のような表記をするしかなかった．

2017 年度の問題も頂点をつなぐ辺のデータを 2 次元で与えており，「どんくり」では前述の Tetra 同様**図 6** のような初期化ができるが，当時の WaPEN では**図 7** のように文字列を分割する形での初期化を使った．

そこで，代入の際にはコンマなどの直後での改行を許すようにした．これにより，値を 2 次元の盤面通りに書き込むことができる（プログラムは図 4 や図 6 とほぼ同じなので省略する）．

ところで，文部科学省の「情報」実践事例集 [21] の「情報 I(3)『コンピュータとプログラミングの実践事例』」でコンウェイのライフゲームが題材として取り上げられているのだが，そのプログラム例ではアクティブなセルを 1 つずつ代入で初期化している（**図 8**）．盤面通りの 2 次元配列による初期化であればグライダーの形を確認しやすいのだが，それを行わないのは後に盤面を大きくするような発展を考えているためである．

ⓒ 2021 Information Processing Society of Japan

```
整数 tate,yoko,Masu[11,9],x,y,nutta,s
tate ← 9
yoko ← 11
Masu[1] ← [0,1,1,1,1,1,1,1,1]
Masu[2] ← [0,1,0,0,0,1,0,0,0,1]
Masu[3] ← [0,1,1,1,0,0,0,1,0,1]
Masu[4] ← [0,9,0,0,0,1,0,1,0,1]
Masu[5] ← [0,1,1,1,0,1,0,1,0,1]
Masu[6] ← [0,1,0,1,1,1,0,1,1,1]
Masu[7] ← [0,1,0,0,0,1,0,1,0,1]
Masu[8] ← [0,1,1,1,0,0,0,1,0,9]
Masu[9] ← [0,1,0,0,0,1,1,1,0,1]
Masu[10]← [0,1,0,1,0,0,0,0,0,1]
Masu[11]← [0,1,1,1,1,1,1,1,1,1]
```

図 5　2018 年度問題の初期化部分（当時の WaPEN 版）

Fig. 5　Initialization part of the problem in 2018 (using WaPEN at that time)

```
tyotensu←8
Hen←{
    {"-","A","B","-","A","-","A","B"},
    {"-","-","-","C","A","-","A","C"},
    {"-","-","-","E","-","E","E","B"},
    {"-","-","-","-","-","E","E","C"},
    {"-","-","-","-","D","A","D"},
    {"-","-","-","-","-","E","D"},
    {"-","-","-","-","-","-","F"}
}
```

図 6　2017 年度問題の初期化部分（どんくり版）

Fig. 6　Initialization part of the problem in 2017 (using Donkuri).

```
整数 tyotensu,hensosu,Siten[22],Syuten[22],kotae,i,j,x,y
文字列 Hen[8,8],Senbun[22],HenData[8]
HenData[1] ← 「-AB-A-AB」
HenData[2] ← 「---CA-AC」
HenData[3] ← 「---E-EEB」
HenData[4] ← 「-----EEC」
HenData[5] ← 「-----DAD」
HenData[6] ← 「------ED」
HenData[7] ← 「-------F」
HenData[8] ← 「--------」
i を 1 から 8 まで 1 ずつ増やしながら,
  j を 1 から 8 まで 1 ずつ増やしながら,
    Hen[i,j] ← substring(HenData[i],j-1,1)
  を繰り返す
を繰り返す
```

図 7　2017 年度問題の初期化部分（当時の WaPEN 版）

Fig. 7　Initialization part of the problem in 2017 (using WaPEN at that time).

```
h=9
w=9
a=np.array([[0]*h]*w)
a[1,2]=1
a[2,3]=1
a[3,1]=1
a[3,2]=1
a[3,3]=1
print(a)
```

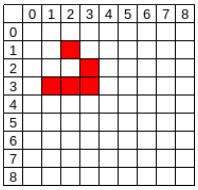

図 8　ライフゲームの盤面初期化

Fig. 8　Initialization of the board of “Conway’s Game of Life”.

4.5　Web ストレージを用いた File I/O（もどき）

PEN には File I/O が実装されており [22], PenFlowchart でもそれをそのまま使うことができるようになっている．しかし WaPEN や PyPEN を開発するにあたってこれを実装してこなかった．Web ブラウザからローカルファイルを操作することができないし，実行にサーバが必要となる実装をしたくなかったためである．

```
(01)Angoubun = ["p","y","e","b",…(省略)…"k","b","d","r","."]
(02)配列 Hindo のすべての要素に 0 を代入する
(03)i を 0 から 要素数(Angoubun)-1 まで 1 ずつ増やしながら:
(04)  bangou = 差分( ケ )          ケ ⓪Angoubun[i]
(05)  もし bangou != -1 ならば:
(06)    コ = コ + 1                コ ④Hindo[bangou]
(07)表示する(Hindo)
```

図 9　共通テスト試作問題の第 5 問

Fig. 9　Question 5 of the prototype problem of the Common examination.

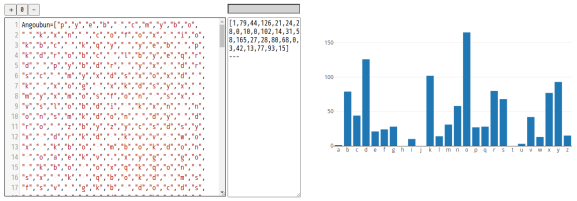

図 10　文字の配列による初期化

Fig. 10　Initialization with an array of characters.

```
tr A-Za-z k-za-za-j < gettysburg.txt > crypt.txt
```

図 11　シーザー暗号で暗号化

Fig. 11　Crypt with Carsar cipher.

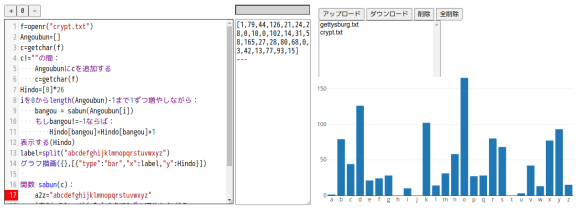

図 12　ファイルによる初期化

Fig. 12　Initialization with a file.

しかし試作問題で出題されたシーザー暗号の問題では，ゲティスバーグ演説全文を暗号化したテキストを 1 文字ずつに分割した配列が初期化に用いられている（**図 9**）．これは 1400 文字以上あるので，4.4 で用いた表記でも何画面にもわたり配列の初期化が続いてしまい，見通しが非常に悪くなってしまう（**図 10**）．むしろ 1 行で書いてしまった方が見やすいかもしれない（数千カラムになるが）．

そこでファイルのような振る舞いをするものを，Web ストレージを用いて実装した．単に Web ストレージの key–value ペアをファイル名とファイルの中身のように扱っているだけではあるが，アップロードされたファイルを読んだり，ファイルに書き込みをしたりするのと同じような振る舞いができている．

この問題の場合**図 11** のように平文の `gettysburg.txt` から，`tr` コマンドを使ってシーザー暗号（鍵は 10）で暗号化したファイル `crypt.txt` を作ってアップロードし，このファイルを 1 文字ずつ読んで配列に追加することで，問題にあるのと同じ配列を生成することができる（**図 12**）．

ⓒ 2021 Information Processing Society of Japan

Vol.2021-CE-162 No.19
Vol.2021-CLE-35 No.19
2021/12/5

5. おわりに

センター試験等のプログラミング問題で省略されてきた初期化部分について，WaPEN や PyPEN に追加実装を行うことによって簡潔に表現できることが確認できた．とはいっても，多くの面では「どんくり」や Tetra に追いつこうとしたということではあるのだが．

以下，DNCL の学習環境に関する私見を述べる．高野らはより自然言語に近いプログラミング環境 Samoyed [23] を開発する中で，DNCL の設計者は DNCL そのものによる学習を良しとしていないのではないかと考察している．実際，大学入試センター試験問題調査官からも，Python や JavaScript などの一般的な言語で演習をすることが望ましいという旨の発言があった [24]．筆者はこれらの考えに反対するものではないが，避けるべきなのは DNCL で演習を行うこと自体ではなく，紙上や限られたことしかできない環境下での演習に終わってしまい，プログラミングを後の学習に利用することにつながらないことではないかと考えている．プログラミングの導入の 1 方法として，あるいはわからなくなったときに振り返る先として，DNCL でのプログラミング学習環境にも意味はあるのではないだろうか．実際 PyPEN に Python のコードが出力できる機能を実装したのは，Python での書き方がわからなくなったときのアンチョコにしてほしいという思いからであった．

参考文献

[1] くじらはんど: なでしこ：日本語プログラミング言語，（オンライン），入手先 ⟨https://www.nadesi.com⟩（参照 2021-10-22）．
[2] くじらはんど: なでしこ 3 大学入試共通テスト手順記述標準言語 DNCL 対応のお知らせ，（オンライン），入手先 ⟨https://www.eznavi.net/site/press/?site=20211013⟩（参照 2021-10-24）．
[3] 大学入試センター：センター試験用手順記述標準言語 (DNCL) の説明，（オンライン），入手先 ⟨https://www.dnc.ac.jp/albums/abm00004841.pdf⟩（参照 2021-10-31）．
[4] 大学入試センター：共通テスト手順記述標準言語 (DNCL) の説明，（オンライン），入手先 ⟨https://www.dnc.ac.jp/albums/abm00040701.pdf⟩（参照 2021-10-31）．
[5] 情報処理学会：大学入学共通テストへの「情報」の出題について，（オンライン），入手先 ⟨https://www.ipsj.or.jp/education/edu202012.html⟩（参照 2021-06-23）．
[6] 大学入試センター: 令和 7 年度以降の試験に向けた検討について—サンプル問題，（オンライン），入手先 ⟨https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7ikou.html⟩（参照 2021-10-31）．
[7] 中村亮太，西田知博，松浦敏雄: プログラミング入門教育用学習環境 PEN，情報処理学会研究報告コンピュータと教育 (CE)，Vol. 2005-CE-081, No. 104 (2005)．
[8] 中西　渉: PenFlowchart の開発，情報処理学会研究報告コンピュータと教育 (CE)，Vol. 2012-CE-113, No. 13 (2012)．
[9] 本多佑希，兼宗　進: ブラウザ上で動作する DNCL 学習環境「どんくり」の開発，情報処理学会研究報告コンピュータと教育 (CE)，Vol. 2018-CE-147, No. 10 (2018)．
[10] 大宮大地，松本嵩大，松浦敏雄，中西通雄: 試験問題作成機能と学習及び受験用機能を持つ DNCL プログラミング環境，情報処理学会研究報告コンピュータと教育 (CE)，Vol. 2019-CE-148, No. 7 (2019)．
[11] 大門　巧，大西建輔: ブラウザ上で動作する DNCL 処理系「Tetra」の開発，情報処理学会研究報告コンピュータと教育 (CE)，Vol. 2019-CE-151, No. 7 (2019)．
[12] 大門　巧，大西建輔，青山　浩: XTetra の開発と授業実践による評価，情報教育シンポジウム，Vol. 2021, No. 36 (2021)．
[13] 中西　渉: WaPEN...DNCL の Web ブラウザ上の実行環境におけるフローチャートなどの実装，情報教育シンポジウム論文集，Vol. 2018, No. 31 (2018)．
[14] 中西　渉: プログラミング学習環境 PyPEN の開発，日本情報科教育学会第 13 回研究報告書，No. 5 (2019)．
[15] 本多佑希，漆原宏丞，兼宗　進: 試験問題記述言語 DNCL の改定に合わせた「どんくり」システムの修正と検討，情報処理学会研究報告コンピュータと教育 (CE)，Vol. 2021-CE-159, No. 2 (2021)．
[16] 下村亮太，中西通雄，松浦敏雄，西田知博，安留誠吾，宮本友介: 初学者用ブロックプログラミング環境 wPEN の改良，情報処理学会研究報告コンピュータと教育 (CE)，Vol. 2021-CE-159, No. 5 (2021)．
[17] 大門　巧: つちのこ，（オンライン），入手先 ⟨https://t-daimon.jp/tsuchinoko/⟩（参照 2021-06-21）．
[18] 情報処理学会情報入試委員会：情報関係基礎アーカイブ，（オンライン），入手先 ⟨https://sites.google.com/a.ipsj.or.jp/ipsjjn/resources/JHK⟩（参照 2021-06-20）．
[19] 本多佑希，長瀧寛之，兼宗　進: DNCL 学習環境「どんくり」，大阪電気通信大学兼宗研究室（オンライン），入手先 ⟨https://dolittle.eplang.jp/dncl⟩（参照 2021-10-31）．
[20] 大門　巧: DNCL 処理系「Tetra」，（オンライン），入手先 ⟨https://sites.google.com/view/tetra-dncl/⟩（参照 2021-10-24）．
[21] 日本産業技術教育学会：高等学校「情報」実践事例集，文部科学省（オンライン），入手先 ⟨https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/mext_01342.html⟩（参照 2021-10-31）．
[22] 西田知博，中村亮太，山本武生，松浦敏雄: プログラミング環境 PEN—描画とファイル I/O 機能の実装，情報教育シンポジウム 2006 論文集，Vol. 2006, No. 8 (2006)．
[23] 高野志歩，田村みゆ，富岡真由，秋信有花，倉光君郎: 擬似コードから考える自然言語を活かしたプログラミング言語，情報教育シンポジウム論文集，Vol. 2021, No. 25, pp. 147–151 (2021)．
[24] 水野修治: 大学入学共通テスト新科目「情報 I」～サンプル問題等とそのねらい～，全国高等学校情報教育研究会第 14 回全国大会（大阪大会）基調講演（オンライン），入手先 ⟨https://www.wakuwaku-catch.net/kouen210801/04/⟩（参照 2021-10-28）．

ⓒ 2021 Information Processing Society of Japan